External Hardware Interrupt

e-Yantra Team
Embedded Real-Time Systems Lab
Indian Institute of Technology-Bombay

IIT Bombay
March 9, 2020

Agenda for Discussion

1. Overview
   - External Interrupt
   - Interrupt Pins
   - Position Encoder
   - Interrupt Calculation
2. Registers
   - SREG
   - EIMSK
   - EICRA
   - EICRB
3. Programs
   - ISR
   - Algorithm for Position Encoder
   - Algorithm for Interrupt Switch

What is an External Hardware Interrupt

What is an External Hardware Interrupt

1. It is an interrupt signal sent to the controller from an external device, like a disk controller or an external peripheral.

What is an External Hardware Interrupt

1. It is an interrupt signal sent to the controller from an external device, like a disk controller or an external peripheral.
2. ATmega2560 has 8 hardware interrupt pins (namely INTn where n can be 0 to 7).

What is an External Hardware Interrupt

1. It is an interrupt signal sent to the controller from an external device, like a disk controller or an external peripheral.
2. ATmega2560 has 8 hardware interrupt pins (namely INTn where n can be 0 to 7).
3. To use an external interrupt, the pin has to be configured as a standard IO input.

What is an External Hardware Interrupt

1. It is an interrupt signal sent to the controller from an external device, like a disk controller or an external peripheral.
2. ATmega2560 has 8 hardware interrupt pins (namely INTn where n can be 0 to 7).
3. To use an external interrupt, the pin has to be configured as a standard IO input.
4. If pin is used as an input, external hardware device can be used to interrupt the controller.

What is an External Hardware Interrupt

1. It is an interrupt signal sent to the controller from an external device, like a disk controller or an external peripheral.
2. ATmega2560 has 8 hardware interrupt pins (namely INTn where n can be 0 to 7).
3. To use an external interrupt, the pin has to be configured as a standard IO input.
4. If pin is used as an input, external hardware device can be used to interrupt the controller.
5. Pin can also be used as an output, but in this case the interrupt is generated by the controller itself.

Interrupt pins

Interrupt pins

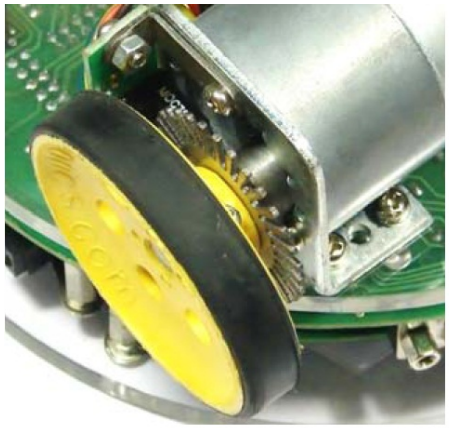
Sr. no	Interrupt	Pin	Firebird V Connection
1	INT0	PD0	-
2	INT1	PD1	-
3	INT2	PD2	-
4	INT3	PD3	-
5	INT4	PE4	Left encoder
6	INT5	PE5	Right encoder
7	INT6	PE6	-
8	INT7	PE7	Interrupt switch

Interrupt pins

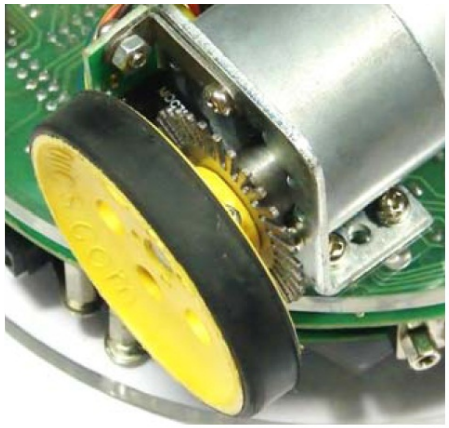
Sr. no	Interrupt	Pin	Firebird V Connection
1	INT0	PD0	-
2	INT1	PD1	-
3	INT2	PD2	-
4	INT3	PD3	-
5	INT4	PE4	Left encoder
6	INT5	PE5	Right encoder
7	INT6	PE6	-
8	INT7	PE7	Interrupt switch

Position encoder

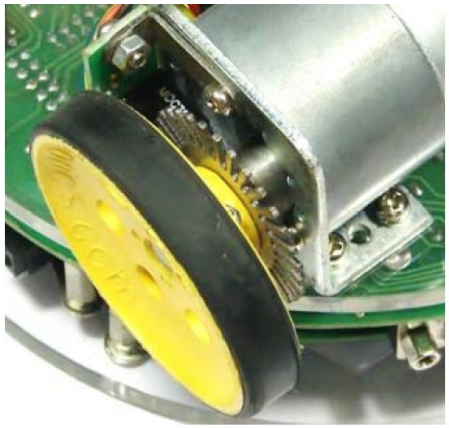
Position encoder



Position encoder

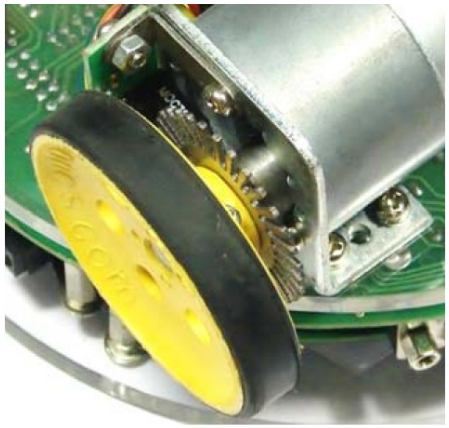


Position encoder



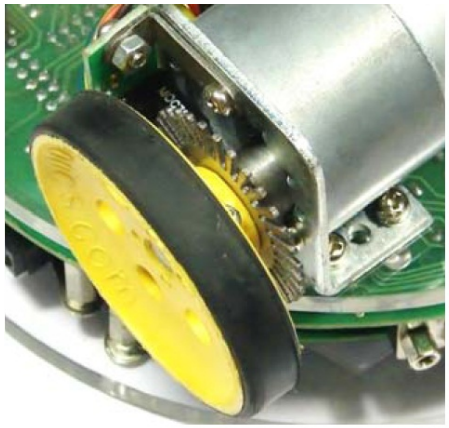
1. Encoder consists of IR LED and photo transistor placed opposite of each other

Position encoder



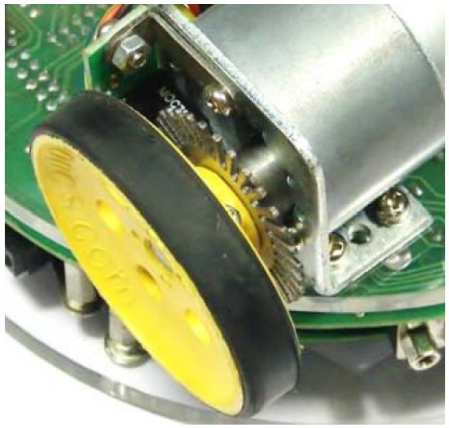
1. Encoder consists of IR LED and photo transistor placed opposite of each other
2. When IR light is interrupted by encoder disc,its output state changes (high to low or low to high)

Position encoder



1. Encoder consists of IR LED and photo transistor placed opposite of each other
2. When IR light is interrupted by encoder disc,its output state changes (high to low or low to high)
3. Output of the encoder is connected to the interrupt pin of the microcontroller

Position encoder



1. Encoder consists of IR LED and photo transistor placed opposite of each other
2. When IR light is interrupted by encoder disc,its output state changes (high to low or low to high)
3. Output of the encoder is connected to the interrupt pin of the microcontroller
4. Left encoder is connected to INT4 and Right encoder is connected to INT5

Some Mathematics...

Some Mathematics...

1. Number of slots in disc = 30

Some Mathematics...

1. Number of slots in disc = 30
2. Number of Pulse/rotation = 30

Some Mathematics...

1. Number of slots in disc = 30
2. Number of Pulse/rotation = 30
3. Diameter of wheel = 52mm

Some Mathematics...

1. Number of slots in disc = 30
2. Number of Pulse/rotation = 30
3. Diameter of wheel = 52mm
4. Resolution of position encoder

Some Mathematics...

1. Number of slots in disc = 30
2. Number of Pulse/rotation = 30
3. Diameter of wheel = 52mm
4. Resolution of position encoder

Some Mathematics...

1. Number of slots in disc $= 30$
2. Number of Pulse/rotation $= 30$
3. Diameter of wheel $= 52mm$
4. Resolution of position encoder

$$= (\pi * d)/30 = 5.44$$

Some Mathematics...

1. Number of slots in disc $= 30$
2. Number of Pulse/rotation $= 30$
3. Diameter of wheel $= 52mm$
4. Resolution of position encoder
   $= (\pi * d)/30 = 5.44$
5. Pulse count
   $=$ distance$/5.44$

SREG- AVR Status Register

This register is used to Globally Enable all Interrupt

SREG- AVR Status Register

This register is used to Globally Enable all Interrupt

Bit	Symbol	Description	Bit Value
7	I	Global Interrupt Enable bit	1
6	T	Bit Copy Storage bit	0
5	H	Half Carry Flag	0
4	S	Sign Bit	0
3	V	Two's Complement Overflow Flag	0
2	N	Negative Flag	0
1	Z	Zero Flag	0
0	C	Carry Flag	0

SREG- AVR Status Register

This register is used to Globally Enable all Interrupt

Bit	Symbol	Description	Bit Value
7	I	Global Interrupt Enable bit	1
6	T	Bit Copy Storage bit	0
5	H	Half Carry Flag	0
4	S	Sign Bit	0
3	V	Two's Complement Overflow Flag	0
2	N	Negative Flag	0
1	Z	Zero Flag	0
0	C	Carry Flag	0

Note: **cli()** and **sei()** are used to clear and set global interrupt respectively

SREG- AVR Status Register

This register is used to Globally Enable all Interrupt

Bit	Symbol	Description	Bit Value
7	I	Global Interrupt Enable bit	1
6	T	Bit Copy Storage bit	0
5	H	Half Carry Flag	0
4	S	Sign Bit	0
3	V	Two's Complement Overflow Flag	0
2	N	Negative Flag	0
1	Z	Zero Flag	0
0	C	Carry Flag	0

Note: **cli()** and **sei()** are used to clear and set global interrupt respectively

(defined in <avr/interrupt.h> header file)

EIMSK- External Interrupt Mask Register

This register is Used to enable Individual External Interrupt

EIMSK- External Interrupt Mask Register

This register is Used to enable Individual External Interrupt

Bit	Symbol	Description	Bit Value
7	INT7	External Interrupt Request 7	0
6	INT6	External Interrupt Request 6	0
5	INT5	External Interrupt Request 5	1
4	INT4	External Interrupt Request 4	1
3	INT3	External Interrupt Request 3	0
2	INT2	External Interrupt Request 2	0
1	INT1	External Interrupt Request 1	0
0	INT0	External Interrupt Request 0	0

EIMSK- External Interrupt Mask Register

This register is Used to enable Individual External Interrupt

Bit	Symbol	Description	Bit Value
7	INT7	External Interrupt Request 7	0
6	INT6	External Interrupt Request 6	0
5	INT5	External Interrupt Request 5	1
4	INT4	External Interrupt Request 4	1
3	INT3	External Interrupt Request 3	0
2	INT2	External Interrupt Request 2	0
1	INT1	External Interrupt Request 1	0
0	INT0	External Interrupt Request 0	0

EIMSK = 0x30

Interrupt Sense Control Bits

Interrupt Sense Control Bits

ISCn1	ISCn0	Description
0	0	The low level of INTn generates an Interrupt request
0	1	Any edge of INTn generates asynchronously an interrupt request
1	0	The falling edge of INTn generates asynchronously an interrupt request
1	1	The rising edge of INTn generates asynchronously an interrupt request

where n = External Interrupt Number (For Atmega2560: n = 0-7)

For External Interrupt = 0
Interrupt Sense Control Bit = ISC01 and ISC00

EICRA- External Interrupt Control Register A

This register is used to select the source to trigger the interrupt

EICRA- External Interrupt Control Register A

This register is used to select the source to trigger the interrupt

Bit	Symbol	Description	Bit Value
7	ISC31	Interrupt Sense control bit for Ext. Interrupt 3	0
6	ISC30	Interrupt Sense control bit for Ext. Interrupt 3	0
5	ISC21	Interrupt Sense control bit for Ext. Interrupt 2	0
4	ISC20	Interrupt Sense control bit for Ext. Interrupt 2	0
3	ISC11	Interrupt Sense control bit for Ext. Interrupt 1	0
2	ISC10	Interrupt Sense control bit for Ext. Interrupt 1	0
1	ISC01	Interrupt Sense control bit for Ext. Interrupt 0	0
0	ISC00	Interrupt Sense control bit for Ext. Interrupt 0	0

EICRA- External Interrupt Control Register A

This register is used to select the source to trigger the interrupt

Bit	Symbol	Description	Bit Value
7	ISC31	Interrupt Sense control bit for Ext. Interrupt 3	0
6	ISC30	Interrupt Sense control bit for Ext. Interrupt 3	0
5	ISC21	Interrupt Sense control bit for Ext. Interrupt 2	0
4	ISC20	Interrupt Sense control bit for Ext. Interrupt 2	0
3	ISC11	Interrupt Sense control bit for Ext. Interrupt 1	0
2	ISC10	Interrupt Sense control bit for Ext. Interrupt 1	0
1	ISC01	Interrupt Sense control bit for Ext. Interrupt 0	0
0	ISC00	Interrupt Sense control bit for Ext. Interrupt 0	0

EICRB- External Interrupt Control Register B

This register is Used to generate Interrupt Signal

EICRB- External Interrupt Control Register B

This register is Used to generate Interrupt Signal

Bit	Symbol	Description	Bit Value
7	ISC71	Interrupt Sense control bit for Ext. Interrupt 7	0
6	ISC70	Interrupt Sense control bit for Ext. Interrupt 7	0
5	ISC61	Interrupt Sense control bit for Ext. Interrupt 6	0
4	ISC60	Interrupt Sense control bit for Ext. Interrupt 6	0
3	ISC51	Interrupt Sense control bit for Ext. Interrupt 5	1
2	ISC50	Interrupt Sense control bit for Ext. Interrupt 5	0
1	ISC41	Interrupt Sense control bit for Ext. Interrupt 4	1
0	ISC40	Interrupt Sense control bit for Ext. Interrupt 4	0

EICRB- External Interrupt Control Register B

This register is Used to generate Interrupt Signal

Bit	Symbol	Description	Bit Value
7	ISC71	Interrupt Sense control bit for Ext. Interrupt 7	0
6	ISC70	Interrupt Sense control bit for Ext. Interrupt 7	0
5	ISC61	Interrupt Sense control bit for Ext. Interrupt 6	0
4	ISC60	Interrupt Sense control bit for Ext. Interrupt 6	0
3	ISC51	Interrupt Sense control bit for Ext. Interrupt 5	1
2	ISC50	Interrupt Sense control bit for Ext. Interrupt 5	0
1	ISC41	Interrupt Sense control bit for Ext. Interrupt 4	1
0	ISC40	Interrupt Sense control bit for Ext. Interrupt 4	0

ISR-Interrupt Service Routine

ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR Format

ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR Format

```
ISR(INTn_vect)
{
    code
}
```

ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR Format

```
ISR(INTn_vect)
{
    code
}
```

where n = External Interrupt Number (For Atmega2560: n=0-7)

Algorithm for Position Encoder

Algorithm for Position Encoder

Problem Statement: Move robot forward for 10cm (1000mm)

Algorithm for Position Encoder

Problem Statement: Move robot forward for 10cm (1000mm)

1. Initialise encoder pins as Input. (PE4 and PE5 as input using DDRE)

Algorithm for Position Encoder

Problem Statement: Move robot forward for 10cm (1000mm)

1. Initialise encoder pins as Input. (PE4 and PE5 as input using DDRE)
2. Initialise External hardware interrupt registers:
   EIMSK = 0x30 - to enable INT4 and INT5
   EICRB = 0x0A - to use falling edge interrupt

Algorithm for Position Encoder

Problem Statement: Move robot forward for 10cm (1000mm)

1. Initialise encoder pins as Input. (PE4 and PE5 as input using DDRE)
2. Initialise External hardware interrupt registers:
   EIMSK = 0x30 - to enable INT4 and INT5
   EICRB = 0x0A - to use falling edge interrupt
3. Enable the global interrupt using sei() function.

Algorithm for Position Encoder

Problem Statement: Move robot forward for 10cm (1000mm)

1. Initialise encoder pins as Input. (PE4 and PE5 as input using DDRE)
2. Initialise External hardware interrupt registers:
   EIMSK = 0x30 - to enable INT4 and INT5
   EICRB = 0x0A - to use falling edge interrupt
3. Enable the global interrupt using sei() function.
4. Move the robot forward. (use necessary functions to configure motor pins)

Algorithm for Position Encoder

Problem Statement: Move robot forward for 10cm (1000mm)

1. Initialise encoder pins as Input. (PE4 and PE5 as input using DDRE)
2. Initialise External hardware interrupt registers:
   EIMSK = 0x30 - to enable INT4 and INT5
   EICRB = 0x0A - to use falling edge interrupt
3. Enable the global interrupt using sei() function.
4. Move the robot forward. (use necessary functions to configure motor pins)
5. Calculate the required pulse count value for 1000 mm.

Algorithm for Position Encoder

Problem Statement: Move robot forward for 10cm (1000mm)

1. Initialise encoder pins as Input. (PE4 and PE5 as input using DDRE)
2. Initialise External hardware interrupt registers:
   EIMSK = 0x30 - to enable INT4 and INT5
   EICRB = 0x0A - to use falling edge interrupt
3. Enable the global interrupt using sei() function.
4. Move the robot forward. (use necessary functions to configure motor pins)
5. Calculate the required pulse count value for 1000 mm.
6. In interrupt service routine, increment the counter value.

Algorithm for Position Encoder

Problem Statement: Move robot forward for 10cm (1000mm)

1. Initialise encoder pins as Input. (PE4 and PE5 as input using DDRE)
2. Initialise External hardware interrupt registers:
   EIMSK = 0x30 - to enable INT4 and INT5
   EICRB = 0x0A - to use falling edge interrupt
3. Enable the global interrupt using sei() function.
4. Move the robot forward. (use necessary functions to configure motor pins)
5. Calculate the required pulse count value for 1000 mm.
6. In interrupt service routine, increment the counter value.
7. Once the pulse count reaches required count, stop the robot.

Algorithm for Interrupt Switch

Algorithm for Interrupt Switch

Problem Statement: Turn On the buzzer whenever the switch is pressed

Algorithm for Interrupt Switch

Problem Statement: Turn On the buzzer whenever the switch is pressed

1. Initialise interrupt switch pin as Input. (PE7 as input using DDRE)

Algorithm for Interrupt Switch

Problem Statement: Turn On the buzzer whenever the switch is pressed

1. Initialise interrupt switch pin as Input. (PE7 as input using DDRE)
2. Initialise External hardware interrupt registers:
   EIMSK = 0x80 - to enable INT7
   EICRB = 0x00 - to use low level interrupt on INT7

Algorithm for Interrupt Switch

Problem Statement: Turn On the buzzer whenever the switch is pressed

1. Initialise interrupt switch pin as Input. (PE7 as input using DDRE)
2. Initialise External hardware interrupt registers:
   EIMSK = 0x80 - to enable INT7
   EICRB = 0x00 - to use low level interrupt on INT7
3. Enable the global interrupt using sei() function.

Algorithm for Interrupt Switch

Problem Statement: Turn On the buzzer whenever the switch is pressed

1. Initialise interrupt switch pin as Input. (PE7 as input using DDRE)
2. Initialise External hardware interrupt registers:
   EIMSK = 0x80 - to enable INT7
   EICRB = 0x00 - to use low level interrupt on INT7
3. Enable the global interrupt using sei() function.
4. In main program, turn OFF the buzzer. (use necessary functions to configure motor pins)

Algorithm for Interrupt Switch

Problem Statement: Turn On the buzzer whenever the switch is pressed

1. Initialise interrupt switch pin as Input. (PE7 as input using DDRE)
2. Initialise External hardware interrupt registers:
   EIMSK = 0x80 - to enable INT7
   EICRB = 0x00 - to use low level interrupt on INT7
3. Enable the global interrupt using sei() function.
4. In main program, turn OFF the buzzer. (use necessary functions to configure motor pins)
5. In interrupt service routine, turn ON the buzzer.

Thank You!